

Deploying context-aware services: A case study of rapid prototyping

Ana M. Bernardos, Paula Tarrío, Josué Iglesias, José R. Casar
{abernardos, paula, josue, jramon}@grpss.ssr.upm.es - Universidad Politécnica de Madrid

Abstract— In this contribution, a real experience of rapid design and deployment of context-aware services for an exhibition hall is detailed. The prototype has been built on a combination of a commercial system (which has been customized and improved to satisfy the prototype needs) with an in-home developed context acquisition framework. In order to partially overcome device fragmentation issues, we have focused on the development of web-based context-aware applications. The whole system has been deployed from scratch under real constraints of time and environment. The objective has been to test the integration problems of context-aware systems, in order to infer some conclusions on what it is needed to generalize them.

Index Terms—Context-aware services, mobile applications.

I. INTRODUCTION

NOWADAYS, many living, working or recreational spaces are equipped with wireless communication infrastructures, sensor networks, radio identification readers or video-cameras. These technologies are currently deployed to offer connectivity, give support to security systems or control ambient parameters, but may also be taken as starting points towards the configuration of smart environments [1]. In parallel, the growing integration of WiFi, Bluetooth, NFC, high-resolution cameras or inertial systems in personal mobile devices is reinforcing their role as user's interfaces to these future highly sensitive environments. As a consequence, traditional mobile services begin to evolve towards personalized context-aware applications, capable of managing the external ambient information and combining it with the particular features and requirements of a target user.

The deployment of context-aware services depends on the development of logical architectures and physical systems capable of acquiring and handling "any information that can be used to characterize the situation of entities (i.e. whether a person, place or object)" [2]. By now, the existing literature gathers a significant number of architecture proposals, many of them claiming the need of decoupling the application development stage from the context provision infrastructures.

Although there are a lot of theoretical works and some testbeds, commercial context-aware applications are far away from generalization. The objective of this contribution is to analyze the complexity of deploying a real system. In this direction, we present an experience of rapid prototyping and deployment of context-aware services, covering from the

identification of blocks of functionalities the system should have and concluding by the integration of reusable components. We also aim at identifying technical and design key aspects that do impact in the user experience.

We have designed and integrated a context-aware system and a bundle of context-based web applications for a fairground. The testbed has been deployed during a real event (a university employment fair).

The paper is organized as follows: Section II gathers some previous experiences in case-study related application areas. Section III presents the boundary conditions of the case-study, with respect to different aspects (such as the user typology or the context description) and Section IV describes the user's context model and the deployed services. The system architecture is described in Section V, under the analysis of the functional framework mentioned above. Lessons learned and final conclusions are covered in Section VI.

II. RELATED RESEARCH

Context-aware systems are able to transparently adapt their response to the user's situation, acquiring and managing information from physical, logical and virtual sensors [3]. More than a decade after the first developments of context-aware systems (starting with the Active Badge project -1992- [4] or the PARCTab project -1994- [5]), a significant number of research initiatives have focused on the development of context-aware software frameworks. Although differing in their architectural approaches (and also in the semantic representation of the context, see [6] for a comparative survey), many of these platforms have been built to simplify the development of context-aware applications, with the objective of making context acquisition, transparent for the applications developer.

In parallel with the evolution of context-aware systems, many areas of application for context-aware mobile services have been explored [7]. In this paper, we present a case-study of context-aware services for fairground-like environments. Many of the services that are useful in this kind of environments are also suitable for extrapolation to other application areas: let us think in the guidance functionalities, location-aware information search engines, resources finders or route optimization services.

With respect to the research and commercial activity focused on context-aware systems or services for fairground or exhibition areas, one of the first trials was built on the well-known Context Toolkit platform: the Conference Assistant [8]. The Conference Assistant is a PC-based application that was configured to provide agenda information, location-aware

This work has been financed by the Spanish Ministry of Education and Science under grant TSI2005-07344 and by the Government of Madrid under grant S-0505/TIC-0255

material downloads and post-event documentation recovery functionalities. The system worked on a WLAN communications infrastructure, using a commercial system for positioning. Also running on a PC, the Hippie prototype [9] provided web information services to prepare, attend and evaluate an event. In Hippie, the user was modelled through a component able to manage preferences, while the position calculated using infrareds, compasses and DGPS.

In 2001, a mobile prototype was proposed by the researchers of the mExpress Project [10] which developed the Wireless Exhibition Guide. This Guide provided information and location services over a positioning platform combining indoor GPS and a Bluetooth cell-id based alternative. In 2003, the SAiMotion project [11] aimed at developing an off-line “nomadic and context-aware” application, which used interactive maps and facilitated the configuration of personalized routes. This system was configured to use a commercial WiFi engine for positioning tasks.

Beyond research-driven experiences, in the last years some technology fairs have made some trials with mobile services. For example, the EGuide product was deployed in 2000 in the German annual exhibition of technology CeBIT [12]. EGuide was built on an infrared positioning system and for the 2001 edition was replaced by the Mobile Fairguide, which used Bluetooth both for positioning and communications. The Mobile Fairguide offered Internet access, proximity events-based information recovery or notification functionalities for the connected devices (PDAs). In 2002, the XGuide was tried. The XGuide application [13] is a guide implemented on the XyberScout platform (a location-aware browser working on a GPS and IrDA infrastructure). In the 2007 and 2008 editions, CeBIT has provided its visitors with a downloadable off-line catalogue of exhibitors, events, facilities and site maps (Mobile Fair Planner), which once installed in the mobile device receives updates and notifications by email. In general, current commercial deployments do not go far beyond the “mobilization” of the traditional exhibitor catalogues.

III. PROTOTYPING CONTEXT-AWARE SERVICES FOR AN EXHIBITION HALL: GENERAL REQUIREMENTS.

In this contribution we refer to a real experience of rapid design and deployment of context-aware services in an exhibition hall. Our testbed has been prepared for the Employment Fair that is annually held in the Telecommunications School of the Technical University of Madrid. Each year, around twenty exhibitors show their corporate activity to the community (students, academicians, researchers and visitors in general), setting their stands for three days. Conferences and parallel recruitment activities are held simultaneously.

From a deeper review of the experiences in Section II, it is possible to state that location technologies are a common weak point in every system, due to the technology they assume in the device side (eg. DGPS needs the devices to be equipped with GPS receivers) or due to their final performance (e.g. IrDA is a line-of-sight dependant technology, Bluetooth cell-

id offers an accuracy of 10 meters and commercial WiFi systems are usually based on unstable propagation models).

Then we present a deployment based on CASanDRA framework, which aim at providing reusable *Context Acquisition Services and Decision and Reasoning Algorithms*. CASanDRA provides different location services, which fulfill different requirements of performance, in order to adapt to the consumer device conditions and to the application needs. It is also possible to interconnect the system to different reasoning engines through its XML based APIs.

On the other hand, previous proposals concentrate on elaborating native wireless guides. In order to cope with device fragmentation and to profit from the growth of semantic web technologies, here we describe a bundle of web-based context-aware services, developed on common web technologies and infrastructure. This approach facilitates the development of services prepared to work before (planning support), during (visit support) and after (memory support) the event.

Our prototype has been elaborated also using adapted commercial off-the-self reasoning technology and it has been completely deployed under real constraints of time and environment.

IV. DESIGNING A PORTFOLIO OF CONTEXT-AWARE SERVICES.

Defining context implies handling personal descriptors, status parameters, temporal data, positioning information and relational records altogether [14]. In our case, the user's context has been described considering personal descriptors like *identity*, *role* (visitor, exhibitor or organizer), *profile* (student, researcher, professor, or practitioner), *preferences* (user's technology interests) and “*social information*” (private groups). These descriptors are provided by the user when registering in the system from his mobile device. Our system also gathers *position* (location information obtained through a positioning system that integrates WiFi and Bluetooth and provides the approximate coordinates of the user's location, translating them into symbolic zones) and *time*. All these descriptors are used as a filter for information about events and related contents. Special communication services are configured among the different groups created. Depending on the information provided, the user will have access to a bundle of services adapted to his personal configuration parameters and his location.

Hierarchical reasoning rules have been built on these information: the first context filter is the symbolic location of the user; afterwards, profile information is applied. The third filter may be the temporal data, the group membership or some personal preferences.

Logically, we consider the fairground space divided into symbolic geometric zones, which are shaped and sized accordingly to the different types of services to be within them. That is to say, location-based and role-adapted bundles of web services will become active and offered to the user when he enters or stays in a given zone. For example, in a conference room, the whole physical space is associated to a zone; when the user enters in, a collection of services will appear in his device: direct accesses to information services, to communication services and to personal applications (Table

1). A zone will be able to contain smaller subzones, each of them representing areas of influence of differentiating elements (e.g. each stand in a fairground may define its own zone of influence covering the physical space it occupies).

As position enables services filtering, we have created and attach the services to physical spaces, in particular to four types of zones: a *General zone*, in which information and communication services by default are provided; the *Entrance zone*, where registration and welcome services are offered; the *Stand zones*, physical influence area of each of the exhibitors, where they will be able to design their own services; and the *Conference room zone*, where information and personal applications' activation will be enabled.

TABLE 1. EXAMPLES OF USER ROLES AND SERVICES.

Visitor	Exhibitor	Organizer
Information services <ul style="list-style-type: none"> -Customized agenda. -Orientation sensitive maps. -Location aware browsing. -Real time guide. -Dynamic planner. -Resources&buddy finders. 	<ul style="list-style-type: none"> -Location-based advertising. -Update of the exhibition geoblog. -Presentations schedule. -Finder. -Same services as a visitor. 	<ul style="list-style-type: none"> -Professional notifications agenda. -Same services as a visitor. - Ambient alerts.
Communication serv. <ul style="list-style-type: none"> -vCards exchange. -VoIP service. -Fair geowiki. -Bookmark storing. -Geo-referenced messaging. -Sending emails or notifications to the exhibitors. 	<ul style="list-style-type: none"> -Ad-hoc communications. -Sending alerts. -Same services as a visitor. 	<ul style="list-style-type: none"> -VoIP service -Same services as a visitor.
Exec. <ul style="list-style-type: none"> -Direct access: notebook, recorder, QR reader, VoIP application. 	<ul style="list-style-type: none"> -Same services as a visitor. 	<ul style="list-style-type: none"> -Same services as a visitor.
Control serv.s <ul style="list-style-type: none"> -Registration, profile management and preferences. -Notifications about activities requiring personal data comm. 	<ul style="list-style-type: none"> -Resources management. -Presence notifications. -Stand visits statistics. 	<ul style="list-style-type: none"> -Resource management. -Tasks assignment. -General statistics. -Alerts on infrastructure problems. -Flow management.

On this approach, we have implemented a bundle of services containing information, communication, execution and control applications (Table 1):

a) Information services, fundamentally designed as web services. They can be triggered manually via a web browser or through a two-dimensional code reading. For visitors, the system includes general information services associated with the activities of the fair (time and position sensitive agenda) and facilities and information services (guided routes and exhibitors localization and finder). Exhibitors can also generate information alerts to the users and the organizers have access to specific schedule.

c) Basic communication services include business virtual cards exchange, activation of a VoIP service (Skype-like), updating the exhibition wiki and storing bookmarks. Visitors can send a notification to an exhibitor if not in its stand, that he will receive in his context-aware client interface in real-time. Users can also leave georeferenced notes to other members in their group, who will find them while passing by the adequated coordinates.

c) Execution services launch device built-in applications, for example, a notebook when the client is in a conference room or a QR code reader when it is known that there are such codes nearby. It is about making more visible those applications that can be useful for the user in a given moment.

d) Control services are configured to let the visitors to manage their personal information and preferences. From the point of view of organizers and exhibitors, control services include access to activity statistics (number of visitors, stay, etc...).

Services are mainly offered as web context-aware applications, and in few cases, as initiators of native applications, previously installed in the PDA. The context-aware client will show to the user a set of access icons (Fig. 1.1) that, when clicked, will forward to the corresponding web service or launch the built-in applications (Fig. 1.2-1.4).

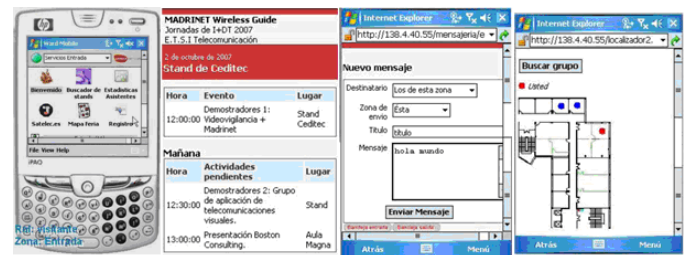


Fig. 1. Some application views: 1) Services offering in a PDA, 2) Web based time-sensitive schedule, 3) Web based georeferenced messaging service, 4) Web based members' group searcher

In the next Section we present the detailed architecture needed to provide these services.

V. SYSTEM'S ARCHITECTURE

When accomplishing the design of the context-aware system for the fairground, we approached it as an integration problem: our objective was to rapidly prototype and deliver a bundle of context aware applications for the event, and to measure the deployment cost in term of workload. For that reason, we did not intend to develop the whole system, but used "off-the-shelf" reasoning technology on the location service of our context acquisition framework CASanDRA, and focused on the development of context-aware web services.

A. CASanDRA acquisition framework.

CASanDRA is a middleware which aims at providing context aware acquisition services and fusion algorithms, isolating the sensor complexity from the application developer. In order to retrieve data samples, CASanDRA offers a set of XML APIs: the application (or the system that provides the services) just has to recover data it needs through HTTP and proceed with a standard parsing.

CASanDRA integrates different services (although only location services have been used for the fairground deployment):

- Location retrieval. By now, five different location services are available: 1) two of them based on device supported WiFi received signal strength fingerprint, 2) one more based on the fusion of WiFi RSS fingerprint and a Bluetooth propagation model, 3) one of them based on the availability of *motes*

(based on ZigBee cell-id) and 4) the latter, based on handling reading notifications of NFC tags placed in significant positions.

- **Movement retrieval.** It allows recovering data from a mobile accelerometer; in particular, we have integrated two types of devices supporting accelerometers: *motest* and a heart monitor with this technology inside.
- **Ambient data retrieval.** Temperature, light, humidity, pressure, etc. sensors connected to a *mote* are easily retrieved through the framework.
- **Biometric data retrieval.** A Bluetooth heart monitor is supported by the framework; from it, it is possible to get heart rate.

The framework, still in evolution, performs data recovery but also integrates different fusion algorithms to provide different qualities of service (for example, in case of positioning), that will be selected depending on the application needs and the available sensing infrastructure. We also aim at integrating services to re-configure sensors, for example to modify the sampling rate or to switch them on/off in order to save power resources.

B. Functional analysis.

There are some basic functional features that should be considered (and solved) when facing the rapid deployment of a context-aware system. Below we comment our approach to each of them, in order to introduce the logical architecture (Fig. 2).

- **Physical data acquisition and context features calculation.** The only physical feature needed for the current deployment was location. As said before, the location service was provided by the acquisition framework CASanDRA, which also translates physical coordinates into its symbolic value (or zones). Among the different available services, we chose to use the WiFi and Bluetooth RSS fingerprint positioning system. This system gets an accuracy of 2.3 meters (details in [15, 16]), enabling users (devices) to locate themselves (and others' devices). The implication from the mobile devices side is the need to run a small application to get WiFi RSS data (that will be subsequently processed in CASanDRA engine).
- **Personal profiles gathering and management.** Knowing the user's role and storing his preferences and group membership was needed to personalize our context-aware web services. On the other hand, in order to make the positioning service work, we needed to store WiFi MAC addresses to identify the devices. Web-based registration applications (running in the Application layer and accessible both from PC and mobile devices) and their connected database supported the process.
- **Object recognition and inference of composite contexts.** Once the basic features are stored or built, the objective is to configure an informational picture combining all the data about a given entity. For the foreground problem, identity (personal and logical), profile (role), preferences, group membership, location and time of the day were the filters that enabled the configuration of a service offering. This was the information on which the Rule-based Decision System had to reason on, supported by the External Context Handler.
- **Decision making.** The system determines which services to offer to the user reasoning on the informational picture

described above. This functionality was implemented in the Services Configuration Manager.

- **Service bundle.** The bundle of services previously described were separately implemented and directly provided by a simple web server (and its connected database), which handed the context-configurable services out to the mobile devices upon the user's request.
- **Authentication and privacy management.** The authentication process was controlled by the Authentication Module, which is connected to the Context Client running in the user's mobile devices. Regarding privacy, this is a horizontal function that should be considered in any stage of the process. In our case, the user was not obliged to provide the system with identity information (pseudonym use is possible), but it was necessary to know some physical information about the mobile device (MACs). With respect to the positioning process, the user is able to control on the client side the service that enables WiFi RSS sending. Secure information storing was also considered.

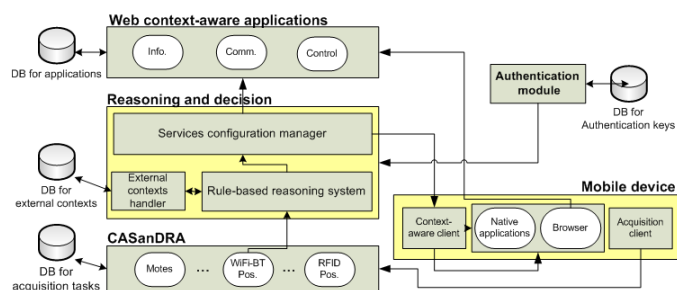


Fig.2. The logical components of the architecture.

C. Logical schema and infrastructure.

The integrated system has already been presented in Fig. 2. The central layer of the system is the reasoning and decision engine, which, on one hand, reasons on the context instance. The context instance is composed with the information provided by the Acquisition Layer and by its External Context Handler. On the other hand, the reasoning and decision engine configures the services offering finally delivered to the Context-aware Client installed in the mobile devices through the Services Configuration Manager.

The External Contexts Handler is an add-on to the off-the-shelf reasoning engine; it facilitates the aggregation of new context parameters, making them available to the Context Provider through XML files. In the device-side, apart from the Context Client, a Positioning Client is needed to make CASanDRA location system work. Some native applications such as a VoIP client or a QR reader have also been installed in the device. Table 2 summarizes the components that have been used to build the system.

As the reader may notice, the different functionalities commented on the previous section are neither linearly implemented nor attached to a single component of the system, but distributed among the different building blocks.

TABLE 2. OFF-THE-SHELF COMPONENTS USED IN THE DEPLOYMENT.

· Rule-based reasoning system	Proprietary system: Appear Context Engine. It allows the integration with external contexts.
· External Contexts Handler	Web server (Apache) with PHP, generates XML understandable by the reasoning system.

· CASanDRA	Application server (Tomcat) with acquisition and fusion algorithms.
· Web applications	Web server (Apache) with PHP. The services use web technologies HTML, CSS, ...
· Acquisition DB	MySQL Data Base
· External Context DB	
· Application Support DB	
· Client	Windows Mobile5.0, WiFi, Bluetooth, camera. Appear Client. Pre-installed native applications (notepad, QR reader, VoIP...).
	CASanDRA's acquisition client.

VI. LESSONS LEARNED AND CONCLUSIONS

From the deployment experience and from the feedback received we can conclude that, in general, the user expects the system to work in a predictable and solid way and does not tolerate irregularities or fall downs. This feeling is highly influenced by the stability and accuracy of the positioning engine the system relies on. Therefore, it is critical to find an effective positioning system in terms of capacity to deal with heterogeneous devices, sufficient accuracy and facility of deployment.

With respect to the last aspect, it is important to bear in mind that many algorithms for indoor positioning perform pattern matching against a stored RSS fingerprint (our positioning algorithm partially does it, although propagation models are also employed). The process of acquiring and storing an RSS fingerprint is a tedious time consuming task, as it is necessary to measure in different points of a virtual grid covering the deployment area. In addition, the RSS fingerprint varies significantly depending on the environment state (people presence, furniture or even ambient conditions). In our case, the exhibition stands were assembled during the day before the event, so the fingerprint had to be updated hastily to adapt it to the new conditions. For these reasons, it is necessary to design algorithms capable of dynamically reconfiguring the RSS fingerprint or even to automatically build it from scratch.

In relation to the user's services, Web context-aware applications are an alternative to traditional native context-aware applications (requiring installation in the mobile device), that minimize the effects of hardware and software fragmentation. Moreover, mobile web browsing experience has greatly improved.

It is also important to notice that using a centralized and externally managed system (both for positioning tasks or context data acquisition) may have some advantages. But it is important to analyze the impact of this strategy on the feeling of control the user should have over the process, as it may be counterproductive. A compromise between user interaction requests, automation and user control is needed.

Although it also depends on the application requirements, our experience shows that it is important to design the system and the services to work with different levels of user's collaboration, regarding the personal information the user has to provide the system with. Privacy requirements may determine the suitability of a positioning solution, but also how to face collaborative architectures (ad-hoc networks) that

may need certain levels of trust to support cooperative processes.

REFERENCES

- [1] M. Weiser. The Computer for the Twenty-First Century. *Scientific American*, Vol. 265. No. 3. Sept. 1991, pp. 94-104.
- [2] A.K. Dey, G. Abowd. Towards a Better Understanding of Context and Context-Awareness Source. *Lecture Notes In Computer Science*; Vol. 1707, pp 304-307, 1999, Karlsruhe, Germany.
- [3] J. Indulska, P. Sutton. Location Management in Pervasive Systems. *Workshop on Wearable, Invisible, Context-Aware, Ambient, Pervasive and Ubiquitous Computing*. 2003, Adelaide, Australia.
- [4] R. Want, A. Hopper, V. Falcao, J. Gibbons. The Active Badge Location System. *ACM Transactions on Information Systems*, Vol. 10, No. 1, Jan. 1992, pp 91-102.
- [5] R. Want, B.N. Schilit, N.I. Adams, R. Gold, K. Petersen, D. Goldberg, J. R. Ellis, M. Weiser. The PARCTAB Ubiquitous Computing Experiment. Technical Report CSL-95-1, Xerox Palo Alto Research Center, March 1995.
- [6] M. Baldauf, S. Dustdar, F. Rosenberg. A Survey on Context-Aware Systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2006, Inderscience Publishers.
- [7] G.Chen, D.Kotz. A Survey of Context-Aware Mobile Computing Research. *Dartmouth Computer Science Technical Report TR2000-381*. 2001.
- [8] A.K. Dey, D. Salber, G.D. Abowd, M. Futakawa. The Conference Assistant: Combining Context-Awareness with Wearable Computing. *ISWC 1999*: 21-28.
- [9] R. Oppermann, M. Specht, I. Jaceniak. Hippie: A Nomadic Information System. *Lecture Notes in Computer Science*, Handheld and Ubiquitous Computing, Volume 1707/1999, Berlin. Editor: Springer.
- [10] I. Mathes, A. Pateli, A. Tsamakov, D. Spinellis. Context aware services in an exhibition environment- the mEXPRESS approach. B. Stanford-Smith et al., editor, *Challenges and Achievements in E-business and E-work: Proc. of the E-business and E-work Conf.*, pages 685-692. IOS Press, October 2002.
- [11] B. Schmidt-Belz, F. Hermann. User Validation of a Nomadic Exhibition Guide. *Lecture Notes in Computer Science*, Mobile Human-Computer Interaction - MobileHCI 2004, Volume 3160/2004, Berlin. Editor: Springer.
- [12] G. Bieber. eGuide Fraunhofer IGD Rostock develops a new generation of electronic exhibition guides. Pressnote, 2000, CeBIT.
- [13] C. Kerer, E. Kirda, C. Krügel. XGuide - A Practical Guide to XML-Based Web Engineering. *Networking Workshops 2002*: 104-117.
- [14] M. Korkea-aho. Context-aware applications survey. Department of Computer Science, Helsinki University of Technology, 2000, <http://www.hut.fi/~mkorkeaa/doc/context-aware.html>.
- [15] E. Collado, A.M. Bernardos, P. Tarrío, J.A. Besada, J.R. Casar. Desarrollo y Despliegue de Sistemas de Localización en WLAN Basados en Huella de Potencia. *Proc. of the 2nd Iberian Conf. on Information Systems and Technologies*, vol. 2, pp. 453-464. June 2007, Porto, Portugal.
- [16] J. Pérez, S. Aparicio, A.M. Bernardos, J.R. Casar, J.R. An Indoor Location System. Based on Bluetooth and WLAN. *Proc. Workshop on User-Centric Technologies and Applications*, pp. 77-86. 2007, Salamanca, Spain.